Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2633528 C 2012/08/07

(11)(21) 2 633 528

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2007/01/18
(87) Date publication PCT/PCT Publication Date: 2007/08/09
(45) Date de délivrance/Issue Date: 2012/08/07
(85) Entrée phase nationale/National Entry: 2008/06/12
(86) N° demande PCT/PCT Application No.: GB 2007/050027
(87) N° publication PCT/PCT Publication No.: 2007/088397
(30) Priorité/Priority: 2006/01/31 (GB0601832.9)

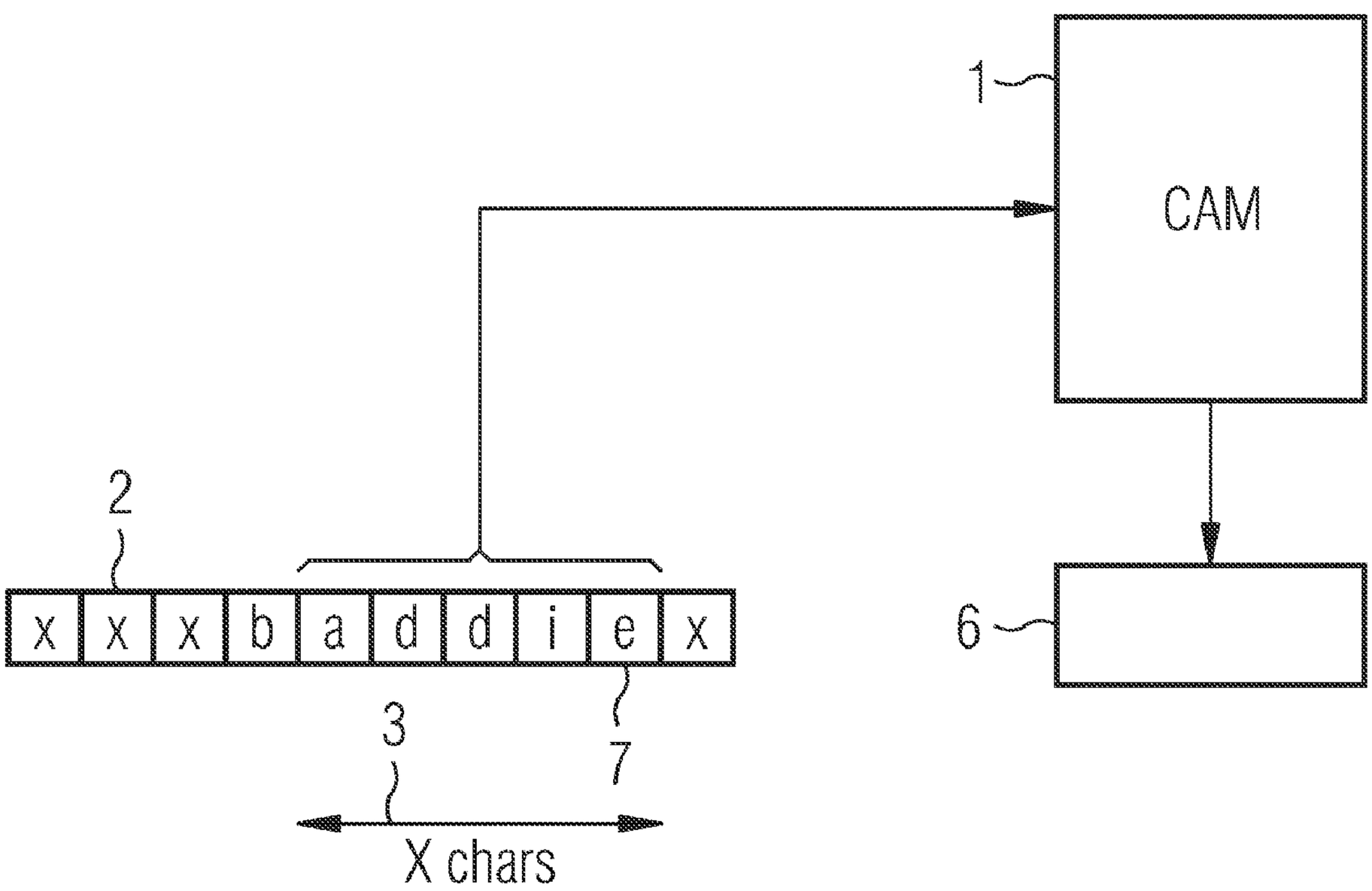
(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01),
G06F 17/30 (2006.01)

(72) Inventeur/Inventor:
DAVIS, SIMON, GB

(73) Propriétaire/Owner:
ROKE MANOR RESEARCH LIMITED, GB

(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : PROCEDE DE FILTRAGE DE TRAFIC A HAUT DEBIT
(54) Title: A METHOD OF FILTERING HIGH DATA RATE TRAFFIC



(57) Abrégé/Abstract:
A method of filtering high data rate traffic (2) based on its content, the method comprising identifying candidate fixed size partial strings (3) within the traffic; comparing characters within the candidate partial string with a content addressable memory (1) containing wanted partial string values and identifying matching traffic; wherein the partial string content includes at least one anchor character (7); wherein the partial string size is set to a predetermined number of characters adjacent to the anchor character; and, wherein partial strings ending in an anchor character are compared with wanted partial string values in the content addressable memory.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 August 2007 (09.08.2007)

PCT

(10) International Publication Number
WO 2007/088397 A3

(51) International Patent Classification:
H04L 29/06 (2006.01) *G06F 17/30* (2006.01)

(21) International Application Number:
PCT/GB2007/050027

(22) International Filing Date: 18 January 2007 (18.01.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0601832.9 31 January 2006 (31.01.2006) GB

(71) Applicant *(for all designated States except US)*: **ROKE MANOR RESEARCH LIMITED** [GB/GB]; Old Salisbury Lane, Romsey Hampshire SO51 0ZN (GB).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **DAVIS, Simon** [GB/GB]; 17 Westering, Romsey Hampshire SO51 7LX (GB).

(74) Agents: **PAYNE, Janice, Julia** et al.; Siemens AG Postfach 22 16 34, 80506 Munich (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

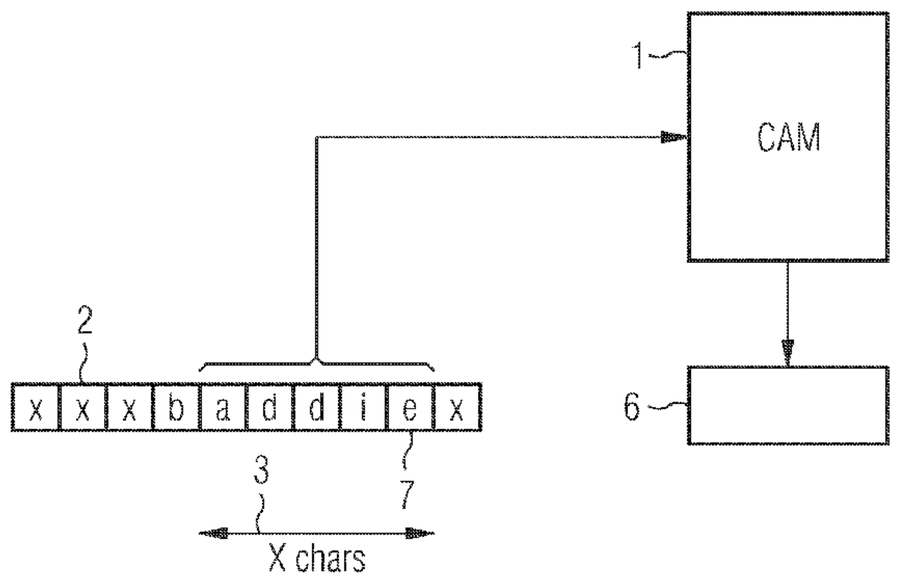
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(88) Date of publication of the international search report:
27 September 2007

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD OF FILTERING HIGH DATA RATE TRAFFIC



(57) Abstract: A method of filtering high data rate traffic (2) based on its content, the method comprising identifying candidate fixed size partial strings (3) within the traffic; comparing characters within the candidate partial string with a content addressable memory (1) containing wanted partial string values and identifying matching traffic; wherein the partial string content includes at least one anchor character (7); wherein the partial string size is set to a predetermined number of characters adjacent to the anchor character; and, wherein partial strings ending in an anchor character are compared with wanted partial string values in the content addressable memory.

WO 2007/088397 A3

A METHOD OF FILTERING HIGH DATA RATE TRAFFIC

This invention relates to a method of filtering high data rate traffic based on its content, in particular in firewalls or for lawful intercept.

There are a number of circumstances in which it is permissible and desirable for a third party to review data traffic before it reaches its final destination. One reason for this is to determine whether there is any improper or damaging content, which the reviewer wishes to exclude from their system, for example, as part of a firewall for a corporate or private network, typically using content based searching or email or internet protocol addresses. Another is in the field of lawful intercept i.e. when law enforcement agencies conduct electronic surveillance of communications, usually approved by the government of the day.

Typically, the data under review is being transmitted over a high bandwidth communication link and the data rates are such that a conventional server or personal computer (pc) cannot search the content at these rates. For example, a pc might have difficulty in operating at more than 1Gbit per second, whereas the communication link may be operating at 10Gbit/s or more.

In accordance with the present invention, a method of filtering high data rate traffic based on its content comprises identifying candidate fixed size partial strings within the traffic; comparing characters within the candidate partial string with a content addressable memory containing wanted partial string values and identifying matching traffic; wherein the partial string content includes at least one anchor character; wherein the partial string size is set to a predetermined number of characters adjacent to the anchor character; and, wherein partial strings ending in an anchor character are compared with wanted partial string values in the content addressable memory.

The present invention cuts down the processing requirement in that only those sections of the data stream for which a specific partial string match, containing a wanted set of partial keywords, is found are forwarded for further processing. The partial strings have a fixed size which is predetermined as part of system analysis, being a trade-off between speed (smaller is better) and false hit probability (larger size leads to less false hits).

The anchor character is typically an essential character, such as an @ in an email address, or a final character in a keyword.

Preferably, a hash function is applied to the partial string to reduce the length of the partial string to be less than or equal to a width of the content addressable memory.

Preferably, padding characters are inserted at either end of the partial string, when the number of characters in the partial string is less than the number of character spaces available in a width of the content addressable memory.

The partial string may be a keyword in a block of text, but preferably, the partial string comprises one of a partial email address, an internet protocol address, a source or destination port number, or other numeric code.

Preferably, matching traffic is forwarded to a secondary processor and store for further processing.

This simplifies the high speed equipment required, as all storage and further processing is done at a lower than real-time data rate, so is less resource intensive. Typically, the secondary processor is a personal computer.

An example of a method of filtering high data rate traffic based on its content in accordance with the present invention will now be described with reference to the accompany drawings in which:

Figure 1 illustrates a first example of the method of the present invention using direct ternary content addressable memory; and,

Figure 2 illustrates a second example of the method of the present invention including a core hash algorithm.

One approach to the problem of lack of processing speed is to filter the traffic of interest based on IP address, for example those of particular email servers, and/or based on port number, to reduce the traffic volume to a level that can be handled by software processes running on a conventional processing platform. The problem with this approach is that line rate filtering is currently relatively simple, so that port numbers, such as for SMTP email protocol, or IP addresses of, for instance, e-mail servers, need to be known in advance. If certain traffic does not use well known ports, or a more generic capability is required, such as searching for a specific word in a particular context in all traffic, rather than searching for a specific

email server, then all packets must be inspected at a line rate which cannot be achieved with software on a general purpose processor. Another problem with needing to know the addresses, or port numbers, is updating that information if the server to which they relate is changed.

The present invention addresses these problems by introducing an algorithm that is split between a programmable front line processor, such as a network processor (NP), at line rate to filter packets and/or sessions that may be of interest and a second line processor, so that the data rate handled by the second line processor is reduced to one slow enough to manage, despite the high data rate of the incoming traffic. Other solutions to this problem either require custom hardware which is expensive and inflexible, or the use of multiple processing platforms to handle the line rate packet processing, which is also expensive.

In the present invention, a high-speed partial string match algorithm is run on the NP in order that the second line processor does not need to handle the same data rates. The NP provides very fast micro-engines that can process packet data, but with limited code and data space, so the algorithm running on the NP needs to be fast and relatively simple.

In a first example of the present invention, as shown in Fig.1, a general string search is carried out, using a direct ternary content addressable memory (TCAM) 1, or network search engine (NSE), look up of a potential key word. A data stream 2 includes a partial string 3 of X characters. This method searches the payload of the data stream 2 character-by-character using a pre-compiled look-up table on each character to determine skip values, based on a target dictionary, as used in well known string search algorithms. Skip values are the number of characters which can be skipped (say Y characters) as no keywords can be matched with the current character in the Y^{th} position.

If a character 7 matches an anchor character, such as the last character of any potential keyword, a look-up of the previous X characters is performed directly using the TCAM functionality of the network search engine NSE. This approach is feasible if the value of X is relatively small so that look-up width can be handled by the TCAM with the resulting table being of sufficient size to hold the dictionary. A hit from the TCAM is then used to filter the packet, or session to an appropriate stream handler on the second line processor 6.

Any keyword can be added to the dictionary including binary sequences. Skip tables based on digrams (two consecutive characters) can also be used to increase skipping efficiency provided that the look-up table can be encoded in the data space available for each micro-engine. For instance it is possible to encode a 2 character skip table as a shorter hierarchical data structure by limiting the number of first characters allowed for the digrams.

Another advantage of partial keyword matching is that any substring of the keyword can be chosen for matching, thus less common character sequences can be chosen to reduce false look-up probability and increase performance.

A second example of the present invention, shown in Fig.2 is described with respect to detection of an e-mail address and includes a core hash algorithm for potential target detection. The workload on the NP micro-engines for checking an email address domain name, is reduced by hashing the characters to the right of the anchor character - here the '@'symbol - and before the next delimiter (invalid character). The NP is provided with a fast hash generator 8 and the resulting hash is compared with known values stored in a table contained by the NSE.

The example of Fig. 2 works as follows. Data from X characters 4 to the left of the '@' to Y characters 5 to the right of the '@' are hashed and looked up against a table of hashes generated from a target ID list stored in the CAM. Provided that the values of X and Y are chosen to guarantee that they are smaller than the smallest size contained in the target e-mail ID list, no real targets will be missed.

Although this example is described with a hash function, the characters can also be presented to the TCAM functionality in the NSE without needing to hash this value, but hashing here has the advantage that the address space can be reduced and therefore the width of table that needs to be stored in the NSE.

For the email example, a large number of hits may occur for particular domains, so at least some portion of the local-part of the e-mail address is usually required, with a target address database stored in the NSE.

It is possible that the local-portion of the e-mail address could be only one character long. If target e-mail addresses contain less than X characters in the local-portion of the address, then special handling is provided. By keeping a check of the last delimiter found whilst searching for the '@' character, the NP code can know the size of the potential local-part of the e-mail address. If this is less than X

characters long, the extra characters are padded with a known value and a hash look-up can be performed to check for this address. As local-part identifiers less than 4 characters long are not common this does not add significant processing overhead.

A further option is that the scan to the right of the @ character is modified to scan until an invalid character, end-of-packet or maximum Y value is encountered. The resultant packet or the whole transmission control protocol (TCP) session to which it belongs is then filtered to a process on the second line processor.

The present invention provides a method for string searching and subsequent filtering within packet data for identifiers of interest, such as e-mail addresses, at multi-gigabit/s line rates using a partial string search technique. A partial string match in a front line processor, such as a network processor or general purpose hardware unit, filters traffic down to a rate that can be handled by software running on a conventional computing platform such as a general purpose server. The filtering can pass through matches deemed to be safe, such as in firewall applications, or those deemed of concern, for lawful intercept. There may be some cases where a match picks up data which is not actually what is being searched for, but by filtering out those which are definitely of no interest, the data rate is brought down to something manageable by the slower, second line processor which can then carry out a finer selection.

The CAM generally only indicates the presence or absence of a match, but in some cases it can store data which is output if a match occurs, such as an index as to which protocol the packet relates to or which process on the second line processor should be used for further processing.

CLAIMS:

1. A method of filtering high data rate traffic based on its content, the method comprising:

identifying candidate fixed size partial strings within the traffic;

comparing characters within a candidate partial string with a content addressable memory containing wanted partial string values and identifying matching traffic;

wherein the candidate partial string content includes at least one anchor character;

wherein a candidate partial string size is set to a predetermined number of characters adjacent to the anchor character; and

wherein partial strings ending in the anchor character are compared with wanted partial string values in the content addressable memory.

2. A method according to claim 1, wherein a hash function is applied to the candidate partial string to reduce the length of the candidate partial string to be less than or equal to a width of the content addressable memory.

3. A method according to claim 1 or 2, wherein padding characters are inserted at either end of the candidate partial string, when the number of characters in the candidate partial string is less than the number of character spaces available in a width of the content addressable memory.

4. A method according to any one of claims 1 to 3, wherein the candidate partial string comprises one of a partial email address, an internet protocol address, a source or destination port number, or a numeric code.

5. A method according to any one of claims 1 to 4, wherein matching traffic is forwarded to a secondary processor and store for further processing.

FIG 1

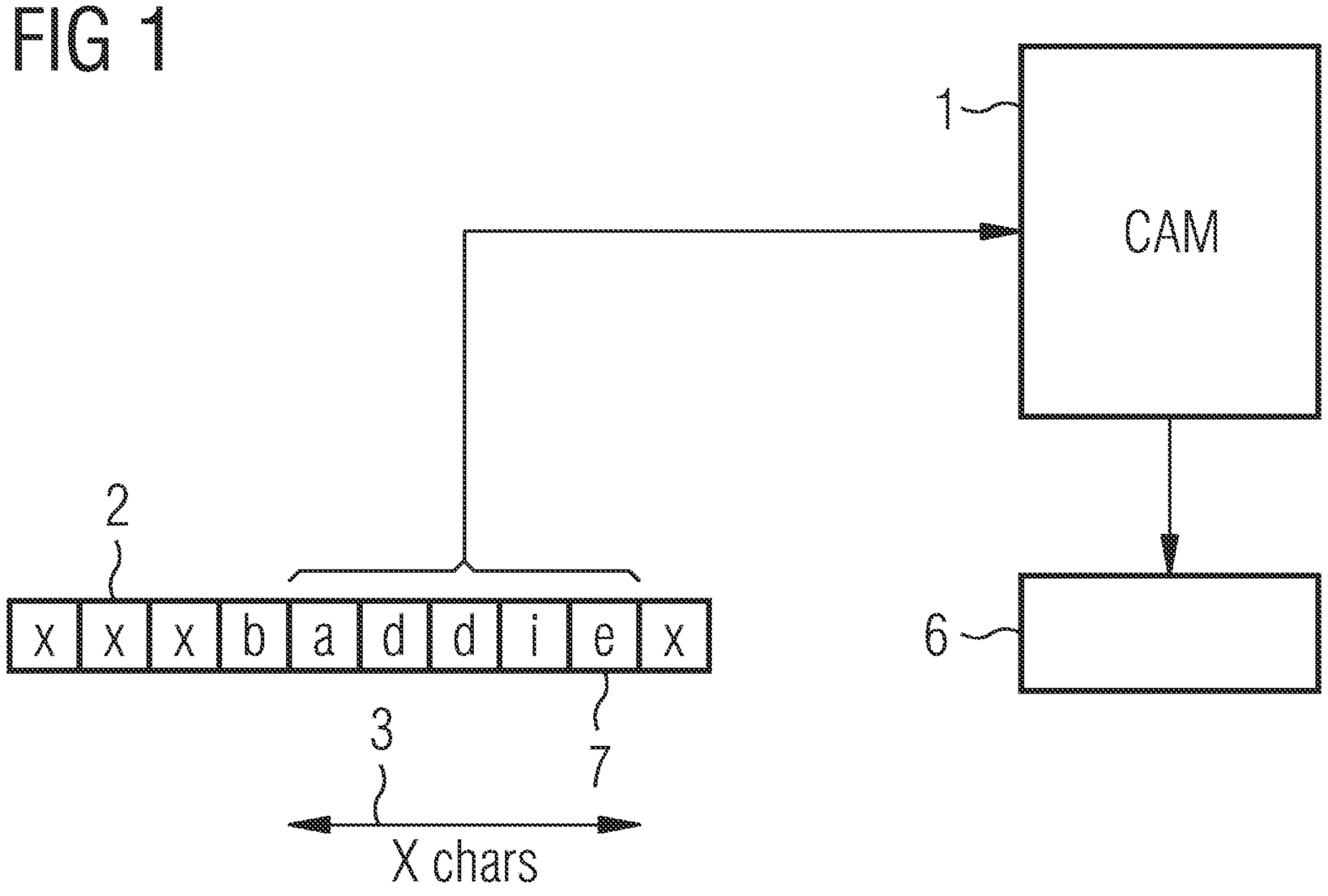
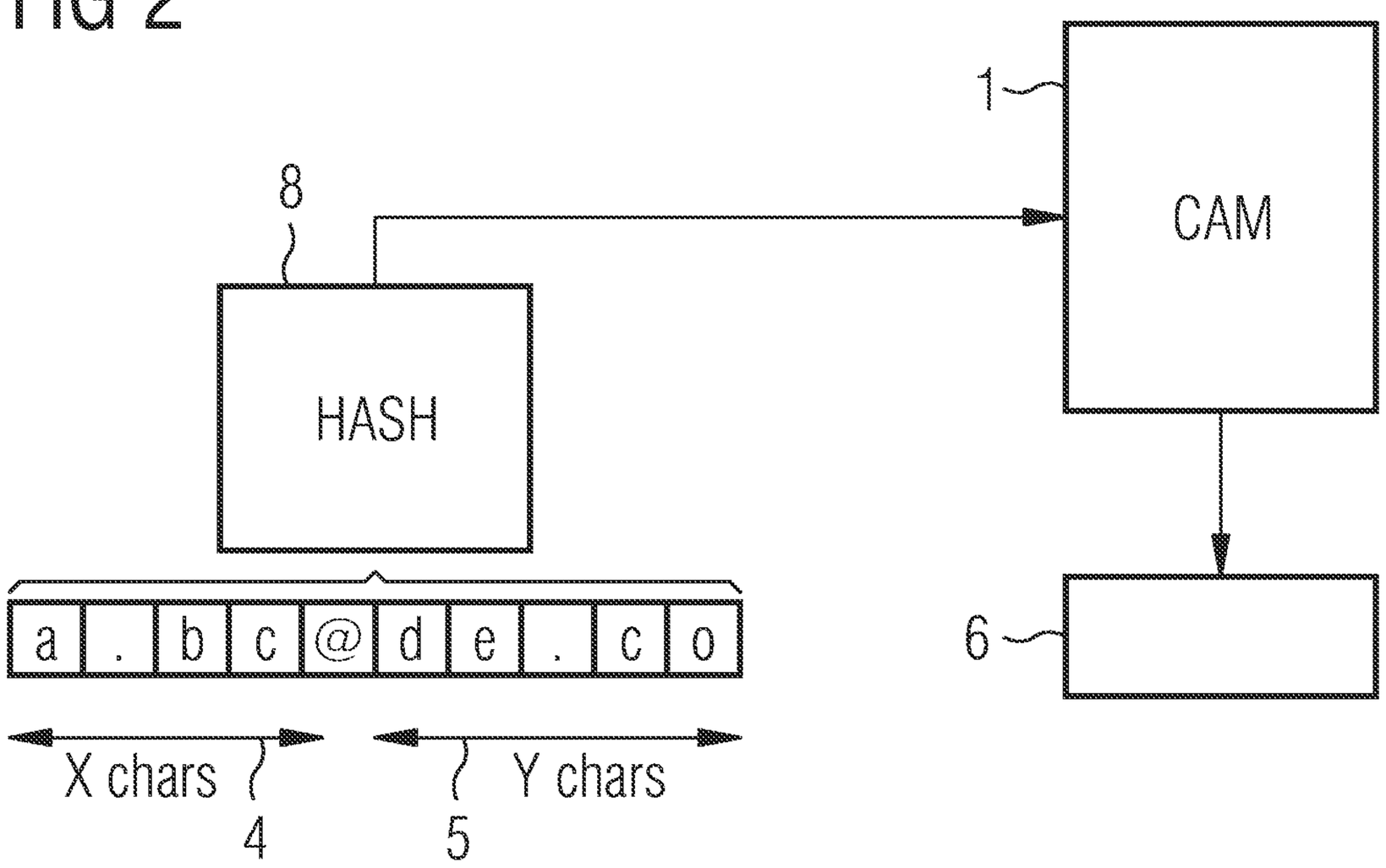
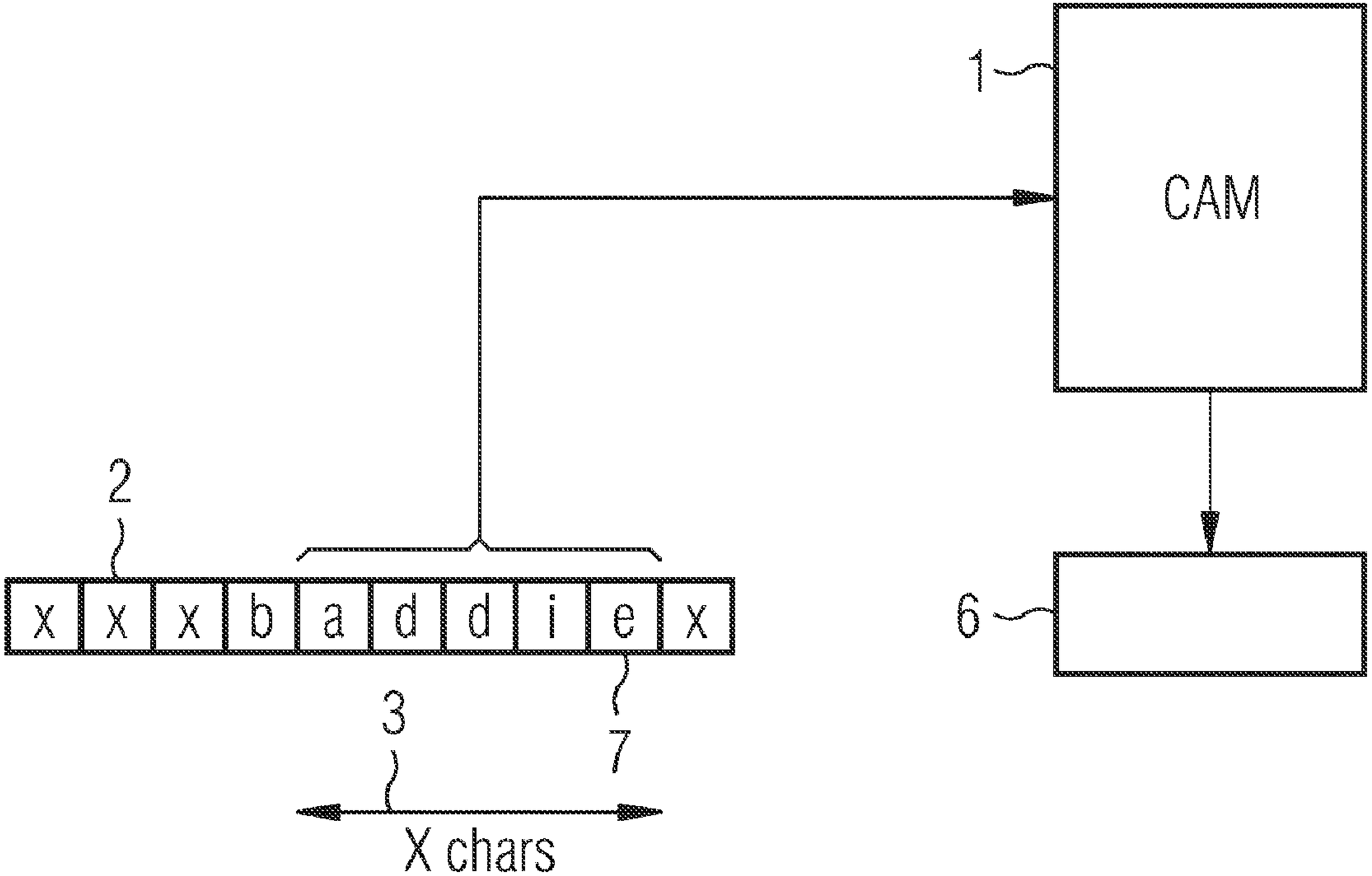

1
CAM
2
x x x b a d d i e x
3
7
X chars
6

FIG 2


1
CAM
8
HASH
a . b c @ d e . c o
6
X chars
4
Y chars
5


1
CAM
2
x x x b a d d i e x
3
7
X chars
6